SECOND WORKSHOP ON SUSTAINABLE SOFTWARE FOR SCIENCE: PRACTICE AND EXPERIENCES (WSSSPE2): SUBMISSION, PEER-REVIEW AND SORTING PROCESS, AND RESULTS

DANIEL S. KATZ[(1)], GABRIELLE ALLEN[(2)], NEIL CHUE HONG[(3)], KAREN CRANSTON[(4)], MANISH PARASHAR[(5)], DAVID PROCTOR[(6)], MATTHEW TURK[(2)], COLIN C. VENTERS[(7)], NANCY WILKINS-DIEHR[(8)]

Abstract. This technical report discusses the submission and peer-review process used by the Second Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2) and the results of that process. It is intended to record both the alternative submission and program organization model used by WSSSPE2 as well as the papers associated with the workshop that resulted from that process.

1. Introduction

The Second Workshop on on Sustainable Software for Science: Practice and Experiences (WSSSPE2)[1] will be held on Sunday, 16 November 2014, in conjunction with the 2014 International Conference for High Performance Computing, Networking, Storage and Analysis (SC14)[2]. WSSSPE2 follows a general initial workshop, WSSSPE1[3] [17, 16], that was held in 2013 in conjunction with the SC13 conference, and a focused workshop, WSSSPE1.1[4], that was held in 2014 in conjunction with the SciPy conference.

Progress in scientific research is dependent on the quality and accessibility of software at all levels and it is critical to address challenges related to the development, deployment, maintenance and overall sustainability of reusable software as well as education around software practices. These challenges can be technological, policy based, organizational, and educational, and are of interest to developers (the software community), users (science disciplines), software engineering researchers, and researchers studying the conduct of science (science of team science, science of organizations, science of science and innovation policy, and social science communities). The WSSSPE1 workshop engaged the broad scientific community to identify challenges and best practices in areas of interest for sustainable scientific software. WSSSPE2 invites the community to propose and discuss specific mechanisms to move towards an imagined future practice for software development and usage in science and engineering. The workshop will include multiple mechanisms for participation, encourage team building around solutions, and identify risky solutions with potentially transformative outcomes. Participation by early career students and postdoctoral researchers is strongly encouraged.

(1) National Science Foundation, Arlington, VA, USA; Computation Institute, University of Chicago & Argonne National Laboratory, Chicago, IL, USA.
(2) University of Illinois, Champaign, IL, USA.
(3) Software Sustainability Institute, University of Edinburgh, Edinburgh, UK.
(4) National Evolutionary Synthesis Center, Durham, NC, USA.
(5) Rutgers Discovery Informatics Institute, Rutgers University, New Brunswick, NJ, USA.
(6) International Consortium of Research Staff Associations, Dublin, Ireland.
(7) University of Huddersfield, School of Computing and Engineering, Huddersfield, UK.
(8) University of California-San Diego, San Diego, CA, USA.

[1] http://wssspe.researchcomputing.org.uk/wssspe2/
[2] http://sc14.supercomputing.org
[3] http://wssspe.researchcomputing.org.uk/wssspe1/
[4] http://wssspe.researchcomputing.org.uk/wssspe1-1/

2. Submissions

The workshop call for papers included the following areas of interest:

- defining software sustainability in the context of science and engineering software
  - how to evaluate software sustainability
- improving the development process that leads to new software
  - methods to develop sustainable software from the outset
  - effective approaches to reusable software created as a by-product of research
  - impact of computer science research on the development of scientific software
- recommendations for the support and maintenance of existing software
  - software engineering best practices
  - governance, business, and sustainability models
  - the role of community software repositories, their operation and sustainability
  - reproducibility, transparency needs that may be unique to science
- successful open source software implementations
  - incentives for using and contributing to open source software
  - transitioning users into contributing developers
- building large and engaged user communities
  - developing strong advocates
  - measurement of usage and impact
- encouraging industry's role in sustainability
  - engagement of industry with volunteer communities
  - incentives for industry
  - incentives for community to contribute to industry-driven projects
- recommending policy changes
  - software credit, attribution, incentive, and reward
  - issues related to multiple organizations and multiple countries, such as intellectual property, licensing, etc.
  - mechanisms and venues for publishing software, and the role of publishers
- improving education and training
  - best practices for providing graduate students and postdoctoral researchers in domain communities with sufficient training in software development
  - novel uses of sustainable software in education (K-20)
  - case studies from students on issues around software development in the undergraduate or graduate curricula
- careers and profession
  - successful examples of career paths for developers
  - institutional changes to support sustainable software such as promotion and tenure metrics, job categories, etc.

Based on the goal of encouraging a wide range of submissions from those involved in software practice, ranging from initial thoughts and partial studies to mature deployments, but focusing on papers that are intended to lead to changes, the organizers wanted to make submission as easy as possible. The call for papers stated:

> We invite short (4-page) **actionable** papers that will lead to improvements for sustainable software science. These papers could be a call to action, or could provide position or experience reports on sustainable software activities. The papers will be used by the organizing committee to design sessions that will be highly interactive

and targeted towards facilitating action. Submitted papers should be archived by a third-party service that provides DOIs. We encourage submitters to license their papers under a Creative Commons license that encourages sharing and remixing, as we will combine ideas (with attribution) into the outcomes of the workshop.

31 submissions were received, and all but one used either arXiv[5] or figshare[6] to self-publish their papers.

3. Peer-Review and Peer-Grouping

The review process was fairly standard, where reviewers bid for papers, then an automated system matched bids to determine assignments, and reviewers then completed their assigned reviews (with an average of 4.9 reviews per paper, and 4.1 reviews per reviewer) This process was done through EasyChair[7], which allowed reviewers to provide scores on relevance and comments to the organizers, which were used to decide which papers to associate with the workshop, and comments to the authors, which were provided back to the authors to allow them to improve their papers.

The organizers decided to list 28 of the papers as significantly contributing to the workshop, a very high acceptance rate, but one that is reasonable, given the goal of broad participation and the fact that the reports were already self-published.

WSSSPE1 was organized into sessions, each of which was aimed at discussing one or more of the themes from the call for papers, with a few paper authors invited to summarize the other papers in that them as a panel, followed by general discussion about that theme. The mapping of papers to themes was done by the organizers.

For WSSSPE2, the organizers wanted to increase the interactivity of the sessions, and to open the process of creating the sessions to the full program committee, the paper authors, and others who might attend the workshop. In order to do this, the organizers decided to use a breakout format for two sessions, and to use an open process to determine the breakout topics. Specifically, well-sorted[8] was used as follows:

(1) Authors were asked to create well-sorted "cards" for the papers. These cards have a title (50 characters maximum) and a body (255 characters maximum).
(2) Authors, program committee members, and members of the WSSSPE mailing list were asked to sort the cards. Each person drags the cards, one by one, into groups. A group can have as many cards as the person wants it to have, and it can have whatever meaning makes sense to that person.
(3) Well-sorted then produces a set of averages of all the sorts, with various numbers of clusters of cards.

The organizers then chose the sort that contained five groups as the one that felt most meaningful, then decided on names for the five groups, namely:

- Exploring Sustainability
- Software Development Experiences
- Credit & Incentives
- Reproducibility & Reuse & Sharing
- Code Testing & Code Review

Finally, since some of the papers were not represented by cards in the process, they were not placed in groups, so the authors of these papers were asked which group seemed the best for their papers, and those papers were then placed in those groups, as listed in the next section of this report.

[5] http://arxiv.org
[6] http://figshare.com
[7] http://easychair.org/
[8] http://www.well-sorted.org

4. Results

The contributed papers that will be discussed at the workshop follow, listed by groups (determined as described in the previous section.)

- Exploring Sustainability
  - Mario Rosado de Souza and Robert Haines and Caroline Jay. Defining Sustainability through Developers' Eyes: Recommendations from an Interview Study [26]
  - Robert Downs, W. Christopher Lenhardt, Erin Robinson, Ethan Davis, Nicholas Weber. Community Recommendations for Sustainable Scientific Software [9]
  - Abani Patra, Matthew Jones, Steven Gallo, Kyle Marcus and Tevfik Kosar. Role of Online Platforms, Communications and Workflows in Developing Sustainable Software for Science Communities [22]
  - Marlon Pierce, Suresh Marru and Chris Mattmann.WSSSPE2: Patching It Up, Pulling It Forward [25]
  - Justin Shi. Seeking the Principles of Sustainable Software Engineering [28]
  - Colin C. Venters, Michael K. Griffiths, Violeta Holmes, Rupert R. Ward and David J. Cooke. The Nebuchadnezzar Effect: Dreaming of Sustainable Software through Sustainable Software Architectures [31]
- Software Development Experiences
  - Jordan Adams, Sai Nudurupati, Nicole Gasparini, Daniel Hobley, Eric Hutton, Gregory Tucker and Erkan Istanbulluoglu. Landlab: Sustainable Software Development in Practice [1]
  - Alice Allen and Judy Schmidt. Looking before leaping: Creating a software registry [2]
  - Carl Boettiger, Ted Hart, Scott Chamberlain and Karthik Ram. Building software, building community: lessons from the ROpenSci project [4]
  - Michael R. Crusoe and C.Titus Brown. Channeling community contributions to scientific software: a hackathon experience [8]
  - Yolanda Gil, Eunyoung Moon and James Howison. No Science Software is an Island: Collaborative Software Development Needs in Geosciences [10]
  - Ted Habermann, Andrew Collette, Steve Vincena, Werner Benger, Jay Jay Billings, Matt Gerring, Konrad Hinsen, Pierre de Buyl, Mark Könnecke, Filipe Rnc Maia and Suren Byna. The Hierarchical Data Format (HDF): A Foundation for Sustainable Data and Software [11]
  - Marcus Hanwell, Patrick O'Leary and Bob O'Bara. Sustainable Software Ecosystems: Software Engineers, Domain Scientists, and Engineers Collaborating for Science [12]
  - Eric Hutton, Mark Piper, Irina Overeem, Albert Kettner and James Syvitski. Building Sustainable Software - The CSDMS Approach [15]
  - W. Christopher Lenhardt, Stanley Ahalt, Matt Jones, J. Aukema, S. Hampton, S. R. Hespanh, R. Idaszak and M. Schildhauer. ISEES-WSSI Lessons for Sustainable Science Software from an Early Career Training Institute on Open Science Synthesis [20]
  - Jory Schossau and Greg Wilson. Which Sustainable Software Practices Do Scientists Find Most Useful? [27]
  - James S. Spencer, Nicholas S. Blunt, William A. Vigor, Fionn D. Malone, W. M. C. Foulkes, James J. Shepherd and Alex J. W. Thom. The Highly Accurate N-DEterminant (HANDE) quantum Monte Carlo project: Open-source stochastic diagonalisation for quantum chemistry [30]
- Credit & Incentives
  - James Howison. Retract bit-rotten publications: Aligning incentives for sustaining scientific software [14]
  - Daniel Katz and Arfon Smith. Implementing Transitive Credit with JSON-LD [18]
  - Ian Kelley. Publish or perish: the credit deficit to making software and generating data [19]

- Reproducibility & Reuse & Sharing
  - Jakob Blomer, Dario Berzano, Predrag Buncic, Ioannis Charalampidis, Gerardo Ganis, George Lestaris and René Meusel. The Need for a Versioned Data Analysis Software Environment [3]
  - Ryan Chamberlain and Jennifer Schommer. Using Docker to Support Reproducible Research [5]
  - Neil Chue Hong. Minimal information for reusable scientific software [13]
  - Tom Crick, Benjamin A. Hall and Samin Ishtiaq. "Can I Implement Your Algorithm?": A Model for Reproducible Research Software [7]
  - Bryan Marker, Don Batory, Field G. Van Zee and Robert van de Geijn. Making Scientific Computing Libraries Forward Compatible [21]
  - Stephen Piccolo. Building Portable Analytical Environments to improve sustainability of computational-analysis pipelines in the sciences [24]
- Code Testing & Code Review
  - Thomas Clune, Michael Rilee and Damian Rouson. Testing as an Essential Process for Developing and Maintaining Scientific Software [6]
  - Marian Petre and Greg Wilson. Code Review For and By Scientists [23]
  - Andrew E. Slaughter, Derek R. Gaston, John Peterson, Cody J. Permann, David Andrs and Jason M. Miller. Continuous Integration for Concurrent MOOSE Framework and Application Development on GitHub [29]

5. Other parts of the workshop

WSSSPE2 will include two keynote presentations, and lightning talks from accepted paper authors, in addition to a set of breakout sessions. The breakouts will allow attendees to participate in discussions about sustainability, future actions, and two of the five areas that resulted from the community submission, review, and grouping process. There will also be reporting from the breakout groups to collect information for all attendees, as well as people who were not able to attend, to understand the discussions.

6. Conclusions

The WSSSPE2 workshop continues our experiment from WSSSPE1 in how we can collaboratively build a workshop agenda. The differences in WSSSPE2 from WSSSPE1 are in using an existing service (EasyChair) to handle submissions and reviews, rather than an ad hoc process, and using an existing service (well-sorted) to allow collaborative grouping of papers into themes by all authors, reviewers, and the community, rather than this being done in an ad hoc manner by the organizers alone.

The fact remains that contributors also want to get credit for their participation in the process. And the workshop organizers still want to make sure that the workshop content and their efforts are recorded. Ideally, there would be a service that would be able to index the contributions to the workshop, serving the authors, the organizers, and the larger community. But since there still isn't such a service today, the workshop organizers are writing this initial report and making use of arXiv as a partial solution to provide a record of the workshop.

After the workshop, one or more additional papers will be created that will include the discussions at the workshop. These papers will likely have many authors, and may be submitted to peer-reviewed journals.

References

[1] Jordan Adams, Sai Nudurupati, Nicole Gasparini, Daniel Hobley, Eric Hutton, Gregory Tucker, and Erkan Istanbulluoglu. Landlab: Sustainable software development in practice. Technical Report 1097629, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1097629.

[2] Alice Allen and Judy Schmidt. Looking before leaping: Creating a software registry. Technical Report 1407.5378, arXiv, 2014. http://arxiv.org/abs/1407.5378.

[3] Jakob Blomer, Dario Berzano, Predrag Buncic, Ioannis Charalampidis, Gerardo Ganis, George Lestaris, and René Meusel. The need for a versioned data analysis software environment. Technical Report 1407.3063, arXiv, 2014. http://arxiv.org/abs/1407.3063.

[4] Carl Boettiger, Ted Hart, Scott Chamberlain, and Karthik Ram. Building software, building community: lessons from the ROpenSci project. Technical Report 1112581, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112581.

[5] Ryan Chamberlain and Jennifer Schommer. Using Docker to support reproducible research. Technical Report 1101910, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1101910.

[6] Thomas Clune, Michael Rilee, and Damian Rouson. Testing as an essential process for developing and maintaining scientific software. Technical Report 1112520, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112520.

[7] Tom Crick, Benjamin A. Hall, and Samin Ishtiaq. "Can I implement your algorithm?": A model for reproducible research software. Technical Report 1407.5981, arXiv, 2014. http://arxiv.org/abs/1407.5981.

[8] Michael R. Crusoe and C.Titus Brown. Channeling community contributions to scientific software: a hackathon experience. Technical Report 1112541, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112541.

[9] Robert Downs, W. Christopher Lenhardt, Erin Robinson, Ethan Davis, and Nicholas Weber. Community recommendations for sustainable scientific software. Technical Report P3VX0DFX, ESIP Commons, 2014. http://dx.doi.org/10.7269/P3VX0DFX.

[10] Yolanda Gil, Eunyoung Moon, and James Howison. No science software is an island: Collaborative software development needs in geosciences. Technical Report 1112561, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112561.

[11] Ted Habermann, Andrew Collette, Steve Vincena, Werner Benger, Jay Jay Billings, Matt Gerring, Konrad Hinsen, Pierre de Buyl, Mark Könnecke, Filipe Rnc Maia, and Suren Byna. The hierarchical data format (HDF): A foundation for sustainable data and software. Technical Report 1112485, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112485.

[12] Marcus Hanwell, Patrick O'Leary, and Bob O'Bara. Sustainable software ecosystems: Software engineers, domain scientists, and engineers collaborating for science. Technical Report 1112482, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112482.

[13] Neil Chue Hong. Minimal information for reusable scientific software. Technical Report 1112528, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112528.

[14] James Howison. Retract bit-rotten publications: Aligning incentives for sustaining scientific software. Technical Report 1111632, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1111632.

[15] Eric Hutton, Mark Piper, Irina Overeem, Albert Kettner, and James Syvitski. Building sustainable software - the CSDMS approach. Technical Report 1407.4106, arXiv, 2014. http://arxiv.org/abs/1407.4106.

[16] Daniel Katz, Sou-Cheng Choi, Hilmar Lapp, Ketan Maheshwari, Frank Löffler, Matthew Turk, Marcus Hanwell, Nancy Wilkins-Diehr, James Hetherington, James Howison, Shel Swenson, Gabrielle Allen, Anne Elster, Bruce Berriman, and Colin Venters. Summary of the first workshop on sustainable software for science: Practice and experiences (WSSSPE1). *Journal of Open Research Software*, 2(1), 2014. http://dx.doi.org/10.5334/jors.an.

[17] Daniel S. Katz, Gabrielle Allen, Neil Chue Hong, Manish Parashar, and David Proctor. First workshop on on sustainable software for science: Practice and experiences (WSSSPE): Submission and peer-review process, and results. Technical Report 1311.3523, arXiv, 2013. http://arxiv.org/abs/1311.3523.

[18] Daniel S. Katz and Arfon M. Smith. Implementing transitive credit with JSON-LD. Technical Report 1407.5117, arXiv, 2014. http://arxiv.org/abs/1407.5117.

[19] Ian Kelley. Publish or perish: the credit deficit to making software and generating data. Technical Report 1112579, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112579.

[20] W. Christopher Lenhardt, Stanley Ahalt, Matt Jones, J. Aukema, S. Hampton, S. R. Hespanh, R. Idaszak, and M. Schildhauer. ISEES-WSSI lessons for sustainable science software from an early career training institute on open science synthesis. Technical Report 1112560, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112560.

[21] Bryan Marker, Don Batory, Field G. Van Zee, and Robert van de Geijn. Making scientific computing libraries forward compatible. Technical Report 1101873, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1101873.

[22] Abani Patra, Matthew Jones, Steven Gallo, Kyle Marcus, and Tevfik Kosar. Role of online platforms, communications and workflows in developing sustainable software for science communities. Technical Report 1112569, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112569.

[23] Marian Petre and Greg Wilson. Code review for and by scientists. Technical Report 1407.5648, arXiv, 2014. http://arxiv.org/abs/1407.5648.

[24] Stephen Piccolo. Building portable analytical environments to improve sustainability of computational-analysis pipelines in the sciences. Technical Report 1112571, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112571.

[25] Marlon Pierce, Suresh Marru, and Chris Mattmann. WSSSPE2: Patching it up, pulling it forward. Technical Report 1112540, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112540.

[26] Mario Rosado de Souza, Robert Haines, and Caroline Jay. Defining sustainability through developers' eyes: Recommendations from an interview study. Technical Report 1111925, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1111925.

[27] Jory Schossau and Greg Wilson. Which sustainable software practices do scientists find most useful? Technical Report 1407.6220, arXiv, 2014. http://arxiv.org/abs/1407.6220.

[28] Justin Shi. Seeking the principles of sustainable software engineering. Technical Report 1405.4464, arXiv, 2014. http://arxiv.org/abs/1405.4464.

[29] Andrew E. Slaughter, Derek R. Gaston, John Peterson, Cody J. Permann, David Andrs, and Jason M. Miller. Continuous integration for concurrent MOOSE framework and application development on GitHub. Technical Report 1112585, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112585.

[30] James S. Spencer, Nicholas S. Blunt, William A. Vigor, Fionn D. Malone, W. M. C. Foulkes, James J. Shepherd, and Alex J. W. Thom. The Highly Accurate N-DEterminant (HANDE) quantum Monte Carlo project: Open-source stochastic diagonalisation for quantum chemistry. Technical Report 1407.5407, arXiv, 2014. http://arxiv.org/abs/1407.5407.

[31] Colin C. Venters, Michael K. Griffiths, Violeta Holmes, Rupert R. Ward, and David J. Cooke. The nebuchadnezzar effect: Dreaming of sustainable software through sustainable software architectures. Technical Report 1112484, figshare, 2014. http://dx.doi.org/10.6084/m9.figshare.1112484.